

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-295246  
 (43)Date of publication of application : 21.10.1994

---

(51)Int.CI. G06F 9/45  
 G06F 9/38

---

(21)Application number : 05-081904 (71)Applicant : TOSHIBA CORP  
 (22)Date of filing : 08.04.1993 (72)Inventor : SAKAI RYUJI  
 TAKEUCHI YOICHIRO

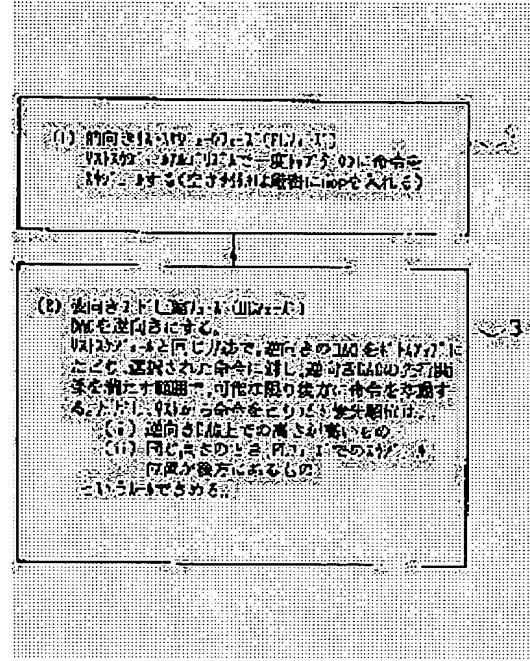
---

## (54) INSTRUCTION SCHEDULING SYSTEM

### (57)Abstract:

**PURPOSE:** To efficiently execute the processing in the instruction scheduling system for executing simultaneously in parallel the instructions.

**CONSTITUTION:** This system is provided with a forward list schedule phase 1 for analyzing a dependent relation of plural instructions and scheduling the instruction by top-down by a graph for showing a precedence relation of the instruction, and a backward code compression phase 3 for analyzing a dependent relation of the instruction concerned subsequently, following the graph by bottom-up by reversing the graph for showing the precedence relation of the instruction and moving the instruction to the utmost within a range for satisfying the precedence relation of the graph being reverse to the selected instruction, thereby scheduling the instruction.



## LEGAL STATUS

[Date of request for examination] 04.03.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-295246

(43)公開日 平成6年(1994)10月21日

(51)IntCl.<sup>5</sup>  
G 0 6 F 9/45  
9/38

識別記号 庁内整理番号  
3 1 0 F 9193-5B  
9292-5B

F I

G 0 6 F 9/ 44

3 2 2 F

技術表示箇所

審査請求 未請求 請求項の数 1 O L (全 6 頁)

(21)出願番号 特願平5-81904

(22)出願日 平成5年(1993)4月8日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 境 隆二

東京都府中市東芝町1番地 株式会社東芝  
府中工場内

(72)発明者 竹内 陽一郎

東京都府中市東芝町1番地 株式会社東芝  
府中工場内

(74)代理人 弁理士 三好 秀和 (外3名)

(54)【発明の名称】 命令スケジューリング方式

(57)【要約】 (修正有)

【目的】 命令を並列に同時実行する命令スケジューリング方式における処理の効率化を図る。

【構成】 複数の命令の依存関係を解析し、命令の先行関係を表すグラフによりトップダウンに命令をスケジュールする前向きリストスケジュールフェーズ1と、次に当該命令の依存関係を解析し、命令の先行関係を表すグラフを逆向きにしてグラフをボトムアップにたどり、選択された命令に対して逆向きのグラフの先行関係を満たす範囲内で可能な限り命令を移動させる後向きコード圧縮フェーズ3とを具備して命令をスケジュールする。

(1) 前向きリストスケジュールフェーズ1 (PLJ<sub>1</sub>-1)  
リストスケジュールフェーズ1で一度トライアルに命令をスケジュールする(空き枠は厳密にnopを入れる)

～1

(2) 後向きコード圧縮フェーズ3 (BCJ<sub>1</sub>-1)  
DAGを逆向きにする。  
リストスケジュールと同じ方法で、逆向きのDAGをボトムアップにたどり、選択された命令に対し、逆向きDAGの先行関係を満たす範囲で、可能な限り後方に命令を移動する。ただし、リストから命令をとりだす優先順位は、  
(i) 逆向きDAG上での高さが高いもの  
(ii) 同じ高さのとき、PLJ<sub>1</sub>-1でのスケジュール位置が後方にあるもの  
という順で並める。

～3

## 【特許請求の範囲】

【請求項1】 複数の命令を並列に同時実行する命令スケジューリング方式において、前記複数の命令の依存関係を解析し、命令の先行関係を表すグラフにより一度トップダウンに命令をスケジュールする前向きリストスケジュールフェーズと、この前向きリストスケジュールフェーズにより一度トップダウンに命令をスケジュールされた後に当該命令の依存関係を解析し、命令の先行関係を表すグラフを逆向きにして当該逆向きのグラフをボトムアップにたどり選択された命令に対して逆向きのグラフの先行関係を満たす範囲内で可能な限り命令を移動させる後向きコード圧縮フェーズとを具備して、

前記前向きリストスケジュールフェーズにより命令をスケジュールした後に当該前向きリストスケジュールフェーズによりスケジュールされたサイクルの範囲内で前記後向きコード圧縮フェーズにより命令をスケジュールする手段と、

を備えたことを特徴とする命令スケジューリング方式。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、複数の命令を並列に同時実行する命令スケジューリング方式のうち、特に、前向きリストスケジュールフェーズによるスケジュールの後に後向きコード圧縮フェーズによりスケジュールすることにより、レジスタの使用量を削減してシステムの処理の遅延を防止し、処理の効率化を図る命令スケジューリング方式に関する。

## 【0002】

【従来の技術】 一般に、複数の命令を同時に並列実行して計算機の処理の高速化を図るために、コンパイラにより命令がスケジュールされてハードウエアが実行するV L I W (Very Long Instruction Word) 方式の計算機がある。上記V L I W方式の計算機においては、複数の命令から次に実行する命令を選択する方式としてリストスケジューリング方式が採用されている。

【0003】 以下、リストスケジューリング方式を述べる。

【0004】 (1) まず、命令の依存関係を解析し、命令をノードとして先行関係をエッジで現すグラフ（以下、DAGという。）を作成する。

【0005】 (2) 上記DAGの各エッジには先行ノードの命令が実行されてから後続ノードが実行可能になるまでの遅延時間を、各ノードにはその命令をスケジュールする際のウェイト（通常DAG上でそのノードの高さ）を付加する。

【0006】 (3) 先行制約のないノードを上記DAGから取り出し、最短実行可能時刻を先行ノードがスケジュールされた時刻と、エッジに付加された遅延時間から求め、当該最短実行可能時刻およびウェイトで優先順位

づけされたリスト（以下、スケジュールリストという。）に追加する。

【0007】 (4) 上記スケジュールリストから順に命令を取り出して当該命令がスケジュール可能、すなわち、現在時刻が最短実行可能時刻より等しいかまたは大きいならばスケジュールする。仮に、スケジュール可能な命令が何もなければ時刻をカウントアップしてリトライする。

【0008】 (5) 以下、全ての命令がスケジュールされるまでステップ(3) からステップ(4) までを繰り返すものであった。

## 【0009】

【発明が解決しようとする課題】 しかしながら、従来の命令スケジューリング方式は、以下に示す問題があった。

【0010】 (1) まず、変数（レジスタ）の使用量が増加するとメモリを使用するため処理の遅延を招来する問題があった。

【0011】 前述したステップ(4) により、リストスケジュールは、そのウェイトに拘らず実行可能な命令ならば当該サイクルにスケジュールする。これにより、リストスケジュールアルゴリズムは、同時実行できる命令の数が有限の場合に対してほぼ最適な解を与える。

【0012】 しかし、基本ブロックが大きくなった場合は、図5に示す配列を加算するときにノード①ではアドレスaを変数t0に代入し、ノード②ではアドレスa+4を変数t0に代入する。以下、ノード③からノード⑦まで実行して配列の加算値Vを演算する。上記配列の加算を前述したリストスケジューリング方式（FL）により実行すると第一のサイクル目にはノード①、②、④用のレジスタを必要とする。また、変数t2は、1サイクル目から3サイクル目まで、変数t3は、2サイクル目から3サイクル目までの3サイクルになり変数の生存範囲が大きくなりレジスタアロケーション時に必要となるレジスタの数が大きくなる傾向がある。上記基本ブロックのサイズが多くなると最悪の場合にはレジスタからデータが溢れてしまいメモリを使用するレジスタスピルが多発する。上記レジスタスピルによりメモリを使用すると当該メモリにロードする時間を要することになり、処理の遅延を招来する問題があった。

## 【0013】

(2) 逆tree形態の依存関係による問題  
上記リストスケジュールは、Tree状の依存関係に対しては処理が効率的に実行されるため、最適になるが逆tree状の依存関係に対しては後続ノード数を、ウェイトの評価に適格に組み入れ無い限り、最適にはならない。また、ウェイトの評価に取り込んだとしても計算コストの増加を招来して、事実上使用ができないくなるものであった。

【0014】 本発明は、このような従来の課題に鑑みて

3

なされたものであり、その目的は、前向きリストスケジュールフェーズによるスケジュールの後に後向きコード圧縮フェーズによりスケジュールすることにより、レジスタの使用量を削減して、システムの処理の遅延を防止し、処理の効率化を図ることによりコンパイラのを迅速に実行するレジスタの使用量を削減してシステムの処理の遅延を防止し、処理の効率化を図る命令スケジューリング方式を提供することにある。

#### 【0015】

【課題を解決するための手段】上記目的を達成するため、本発明は、複数の命令を並列に同時に実行する命令スケジューリング方式において、前記複数の命令の依存関係を解析し、命令の先行関係を表すグラフにより一度トップダウンに命令をスケジュールする前向きリストスケジュールフェーズと、この前向きリストスケジュールフェーズにより一度トップダウンに命令をスケジュールされた後に当該命令の依存関係を解析し、命令の先行関係を表すグラフを逆向きにして当該逆向きのグラフをボトムアップにたどり選択された命令に対して逆向きのグラフの先行関係を満たす範囲内で可能な限り命令を移動させる後向きコード圧縮フェーズとを備えて、前記前向きリストスケジュールフェーズにより命令をスケジュールした後に当該前向きリストスケジュールフェーズによりスケジュールされたサイクルの範囲内で前記後向きコード圧縮フェーズにより命令をスケジュールする手段とを備えたことを要旨とする。

#### 【0016】

【作用】上述の如く構成すれば、まず、複数の命令の依存関係を解析し、命令の先行関係を表すグラフにより一度トップダウンに命令をスケジュールする前向きリストスケジュールフェーズにより命令をスケジュールする。上記前向きリストスケジュールフェーズにより命令をスケジュール後に後向きコード圧縮フェーズにより当該前向きリストスケジュールフェーズによりスケジュールされたサイクルの範囲内で当該命令の依存関係を解析し、命令の先行関係を表すグラフを逆向きにして当該逆向きのグラフをボトムアップにたどり選択された命令に対して逆向きのグラフの先行関係を満たす範囲内で可能な限り命令を移動させて命令をスケジュールするので、レジスタの使用量を削減して、システムの処理の遅延を防止できる。

#### 【0017】

【実施例】以下、本発明の一実施例を図面に基づいて説明する。

【0018】図1は本発明の命令スケジューリング方式の一実施例に係る機能を示すブロック図である。

【0019】上記命令スケジューリング方式は、前向きリストスケジュールフェーズ1および後向きコード圧縮フェーズ3とを備えて、オペレーションシステム（以下、OSという。）による計算機システムの資源を効率

4

的に使用し、スループットを高めて応答時間を短縮するために用いられるものである。

【0020】上記前向きリストスケジュール（Forward List schedule）フェーズ（以下、FLフェーズという。）1は、

（1）まず、命令の依存関係を解析し、命令をノードとして先行関係をエッジで現すグラフ（以下、DAGという。）を作成する。

【0021】（2）上記DAGの各エッジには先行ノード10の命令が実行されてから後続ノードが実行可能になるまでの遅延時間を、各ノードにはその命令をスケジュールする際のウェイト（通常DAG上のそのノードの高さ）を付加する。

【0022】（3）先行制約のないノードを上記DAGから取り出して最短実行可能時刻を先行ノードがスケジュールされた時刻と、エッジに付加された遅延時間から求め、当該最短実行可能時刻およびウェイトで優先順位づけされたリスト（以下、スケジュールリストという。）に追加する。

【0023】（4）上記スケジュールリストから順に命令を取り出して当該命令がスケジュール可能、すなわち、現在時刻が最短実行可能時刻より等しいかまたは大きいならばスケジュールする。仮に、スケジュール可能な命令が何もなければ時刻をカウントアップしてリトライする。

【0024】（5）以下、全ての命令がスケジュールされるまでステップ（3）からステップ（4）までを繰り返すものであった。

【0025】上記ステップ（1）から（5）に示されるリストスケジュールアルゴリズムで一度トップダウンに命令をスケジュールする。なお、空きサイクルは場合には厳密にnopをいれる。

【0026】後向きコード圧縮（Backward Compaction）フェーズ（以下、BCフェーズという。）3は、FLフェーズ1により作成されたDAGを逆向きにし、リストスケジュールを同じ方法により逆向きのDAGをボトムアップにたどり、選択された命令に対して逆向きDAGの先行関係を満たす範囲内で可能な限り後方に命令を移動する。

【0027】ただし、リストから命令を取り出す優先順位は、

①逆向きDAG上の高さの高いもの

②同じ高さのときFLフェーズでのスケジュール位置が後方にあるものというルールで決める。

【0028】なお、②の条件は、命令が移動した際に生じる命令の空きに後続の命令を移動させてなるべく有効利用できるようにすることを狙っている。

【0029】次に、前述したFLフェーズ1およびBCフェーズ3を用いて命令をスケジュールする動作を図250および図3を用いて説明する。

【0030】まず、図2の命令をノードとして先行関係をエッジで現すグラフDAGに示す逆tree形状の命令の先行関係のないノード①およびノード②は、高さとともに2であり、エッジが1である。また、先行関係のないノード③は、高さが零であり、エッジが1である。上記ノード①およびノード②に依存するノード⑤は高さが1で、エッジが1である。上記ノード⑤およびノード③に依存するノード⑨は、高さが零である。

【0031】一方、tree形状の先行関係のないノード④は、高さが1でありエッジ1によりノード⑥、ノード⑦およびノード⑧が依存している。上記ノード⑥、ノード⑦およびノード⑧の高さは、すべて零である。

【0032】以上のDAGをFLフェーズ1によりノードをスケジュールすると1サイクル目には先行関係のないノード①、ノード②およびノード③が配置される。2サイクル目には高さ1のノード⑤およびノード④が、3サイクル目には高さ零のノード⑨、ノード⑥およびノード⑦が、4サイクル目には高さ零のノード⑧が配置されるスケジュールリスト5が作成される。

【0033】なお、FLフェーズ1ではノードの高さの大きいノードから取り出す。すなわち、高さの大きいノードから処理されることになる。

【0034】次に、FLフェーズ1の後にBCフェーズ3によりスケジュールするとノードの配置は、前述した如くFLフェーズ1に比べて高さの計算が逆になる。即ち、図3に示す如く先行関係のないノード①からノード④の高さは零になり、ノード⑤からノード⑧の高さは1になり、ノード⑨の高さは2になる。なお、エッジの長さは1である。

【0035】上記BCフェーズ3は、FLフェーズ1で得たクリティカルパス長を越えない範囲内で変数の生存範囲を大幅に短縮する。また、BCフェーズ3は、高さの高いノード⑨を図2に示すスケジュールリスト7の4サイクル目にスケジュールして、次に、高さ1のノード⑤からノード⑦のうちノード⑥を4サイクル目にスケジュールする。さらに、BCフェーズ3は、高さが1のノード⑤と高さが零のノード③を2サイクル目にスケジュールして、高さ零のノード①およびノード②を2サイクル目にスケジュールする。したがって、FLフェーズ1により上のサイクルに上がり過ぎた命令は、BCフェーズ3で変数の生存範囲が可能な限り最小になるように再配置される。

【0036】更に、FLフェーズ1によりスケジュールされたデータ構造を図4に示す。

【0037】数バイトのデータがポインタにより結ばれており第一のキューにはノード①②③が結ばれて、第二のキューにはノード⑤、ノード④が、第三のキューには

ノード⑨、ノード⑥およびノード⑦が、そして、第四のキューにはノード⑧が接続されている。

【0038】これにより、図5に示す配列の加算の場合には、FLフェーズ1(FL)によりスケジュールした後にBCフェーズ3(BC)によりスケジュールするため、FLにおいて1サイクル目ではノード①②③の3つのレジスタを必要とするのに対し、BCフェーズ3のスケジュールによりノード①②の2つのレジスタで済む。また、変数t2の生存範囲もFLフェーズ1のスケジュールでは1サイクルから3サイクルであるのに対し、BCフェーズ3のスケジュールにより2サイクルから3サイクルまでの2サイクルで済むことになる。

【0039】従って、関数の出入り口でのレジスタの対比、復元が高速に行うことができるとともに、レジスタスピンコードの発生頻度が少くなり、レジスタの使用量を削減できる。

【0040】また、目的コードのパス長の短い命令スケジュールを実現できるため、コンパイラを生成した後に実行する時間を短縮できる。

【0041】更に、FLフェーズでは最適になりにくい逆tree依存関係を含むケースの実行サイクルを改善する。

【0042】したがって、命令を並列に実行するシステムの処理時間を迅速にする。

#### 【0043】

【発明の効果】以上説明したように、本発明では、前向きリストスケジュールフェーズにより命令をスケジュール後に後向きコード圧縮フェーズにより命令をスケジュールするので、前向きリストスケジュールフェーズによるスケジュールの後に後向きコード圧縮フェーズによりスケジュールすることにより、レジスタの使用量を削減してシステムの処理の遅延を防止し、処理の効率化を図るレジスタの使用量を削減して、システムの処理の遅延を防止し、処理の効率化を図ることによりコンパイラの迅速な実行を実現できる。

#### 【図面の簡単な説明】

【図1】本発明の命令スケジューリング方式の一実施例に係る機能を示すブロック図である。

【図2】グラフおよびスケジュールリストを示す図である。

【図3】グラフを示す図である。

【図4】データ構造を示す図である。

【図5】配列の加算を示す図である。

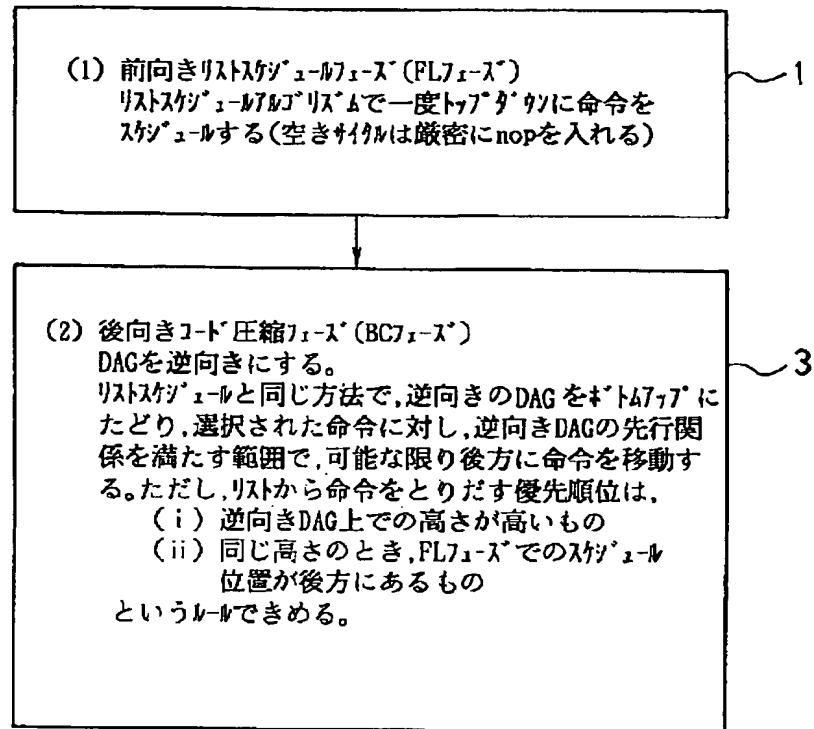
#### 【符号の説明】

1 FLフェーズ

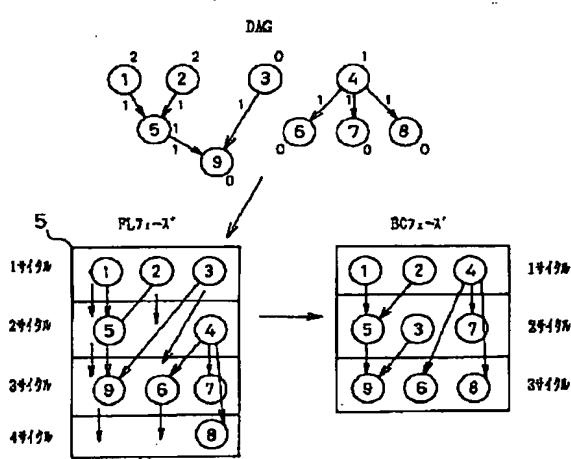
3 BCフェーズ

5, 7 スケジュールリスト

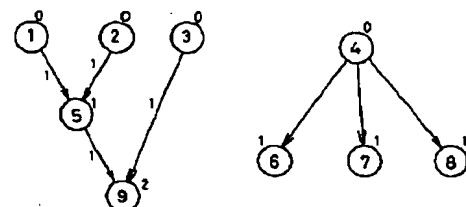
【図1】



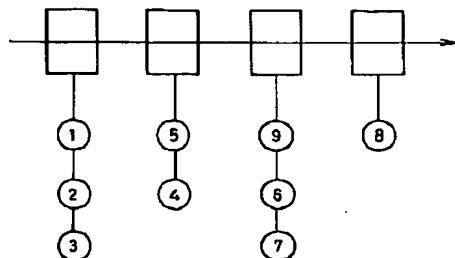
【図2】



【図3】



【図4】



【図5】

$$V = a[0] + a[1] + a[2] + a[3] :$$

